

AI-based actuator/sensor fault detection with low computational cost for industrial applications

Konstantinos Michail, *Member, IEEE*, Kyriakos M. Deliparaschos, Spyros G. Tzafestas, *Life Fellow, IEEE*, and Argyrios C. Zolotas, *Senior member, IEEE*

Abstract—A low computational cost method is proposed for detecting actuator/sensor faults. Typical model-based fault detection units for multiple sensor faults, require a bank of estimators (i.e., conventional Kalman estimators or artificial intelligence based ones). The proposed fault detection scheme uses an artificial intelligence approach for developing of a low computational power fault detection unit abbreviated as ‘*i*FD’. In contrast to the bank-of-estimators approach, the proposed *i*FD unit employs a single estimator for multiple actuator/sensor fault detection. The efficacy of the proposed fault detection scheme is illustrated through a rigorous analysis of the results for a number of sensor fault scenarios on an electromagnetic suspension system.

Index Terms—fault tolerant control, actuator/sensor fault detection, reconfigurable control, loop-shaping robust control design, electromagnetic suspension, maglev trains, neural networks, artificial intelligence.

I. INTRODUCTION

Design of fault tolerant control systems involves a trade-off between system economic aspects, control performance and reliability. Reliability is important in control systems particularly to safety-critical systems. In many cases such as Unmanned Area Vehicles (UAVs), control methods that guarantee control reliability increase the computational cost of the already limited system resources [1], [2].

Fault accommodation in modern control systems can be achieved by either Passive Fault Tolerant Control (PFTC), or Active FTC (AFTC), or a combination of both [3]. In order to ensure controller robustness in the PFTC, prior knowledge of the faults is necessary. In the AFTC method, the approach used in this paper, a reconfigurable controller is designed in combination with a Fault Detection and Isolation (FDI) unit. Although the complexity of the design is increased, the computational cost is reduced and the performance is guaranteed under faults [4].

Fault tolerance against sensor (and/or actuator) failures is extensively considered in the literature using both model-based and model-free methods. This is because the stability and performance of the plant under control depends on them and because failures on those components occur on a relatively regular basis [5]–[7]. Consequently, the FDI unit must be able to detect and isolate any faulty sensors (and/or actuators)

Konstantinos Michail is with the Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, Limassol, CY and SignalGeneriX Ltd, Limassol, CY (e-mail: kon_michael@ieee.org).

Kyriakos M. Deliparaschos is with the Electrical and Computer Engineering Department and Informatics, Cyprus University of Technology, Limassol, CY and CTVR, Trinity College, University of Dublin, IR (email: k.deliparaschos@cut.ac.cy).

Spyros G. Tzafestas is with the School of Electrical and Computer Engineering, National Technical University of Athens, GR (email: tzafesta@softlab.ntua.gr).

Argyrios C. Zolotas is with the College of Science, School of Engineering, University of Lincoln, Lincoln, UK (email: azolotas@lincoln.ac.uk).

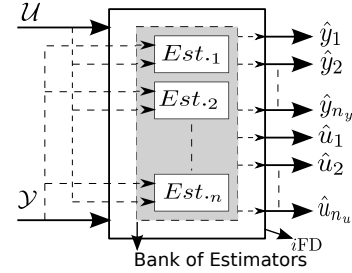


Fig. 1. Actuator/Sensor signals estimation using bank of estimators (dotted lines) vs the proposed estimator (bold lines).

based on their readings, and then the controller must be reconfigured to maintain the performance using the remaining healthy sensors (and/or actuators).

In relation to sensor fault tolerance, a few FDI methods exist such as bank of Neural Networks (NNs) or Kalman Estimators (KE) [1], [8], or other methods that use the information from the remaining healthy sensors to reconstruct the lost signal [9]. Both of the FDI approaches mentioned above are useful since sensor redundancy is avoided, although the required computational power is usually higher due to multiple estimators that are used.

In contrast to the KE approach, NNs present an increased False Alarm Rates (FAR) mainly due a very small residual that remains after fault estimation [9]. Nevertheless, NNs are widely used since they can work without having precise and formal knowledge of the system [10]. In situations where many sensors (actuators) exist, either of the two aforementioned approaches require a co-working bank of parallel estimators (shown with dotted lines in Fig. 1) for multiple faults detection [11]. Consider the example of one actuator and n_y the total number of sensors and assume that not all of them can fail, then the resulting number of sensor fault combinations is $2^{n_y} - 1$. In order to detect the sensor faults in the previous example, an equal number of estimators is required which consequently increases both the complexity of the control design and the computational resources (since a number of estimators in parallel is needed). NNs have been widely used in FTC systems, specifically in Fault Detection (FD) methods and in sensor FDI [9], [12].

In this work the authors propose a low computational cost FD mechanism (*i*FD) based on Artificial Intelligence (AI) methods, that reduces the complexity and computational power of a bank-based FD. More specifically, in Fig. 1 a single estimator (solid line box) replaces a bank of estimators that work in parallel (dotted lines). The single estimator achieves the same performance as the bank-based FD but with substantially lower computational cost. The methodology presented here is a substantial extension of the authors' work in [13], [14]. The proposed framework is tested on an Electro-

Magnetic Suspension (EMS) system used on maglev trains to support the mass of the vehicle and passengers as well as to ensure proper ride quality during the journeys [15].

The proposed FD scheme is also compared with a bank of Kalman Estimators. Since the introduction of the basic linear KE, several versions of the KE have been developed mainly for non-linear systems [16]. Although practically applicable, they are computationally expensive, hence the linear KE approach is used for the comparison in order to have the lowest computational cost possible.

The remainder of the paper is organized as follows: In Section II we provide an outline of the proposed *i*FD approach. Section III presents modeling issues for the EMS system. Section IV validates the efficacy of the proposed method based on the results from a number of simulation runs for various fault scenarios. Finally, in Section V we conclude the paper.

II. THE PROPOSED FAULT DETECTION SCHEME

The proposed FD scheme is illustrated in Fig 2a. A typical industrial system has a set of inputs (control signals) and outputs (measurement signals). In practice, the inputs are driven by a set of actuators, \mathcal{U} , and the outputs are measured by a set of sensors, \mathcal{Y} . The control and measurement signals refer to actuators and sensors respectively.

When one or more sensors (and/or actuators) are impaired those signals are distorted, effectively leading to performance degradation or even instability of the closed-loop. The sets of actuators and sensors are defined as $\mathcal{U} = [u_1, u_2, \dots, u_{n_u}]$ and $\mathcal{Y} = [y_1, y_2, \dots, y_{n_y}]$, where u_j is the j^{th} actuator, y_j is the j^{th} sensor and n_u, n_y are the total number of actuators and sensors respectively.

The control loop features a bank of controllers $[K_1, K_2, \dots, K_{n_{yu}}]$ and two isolation units that are responsible for isolating faulty actuator and sensor signals when these occur. The *i*FD mechanism comprises a NN-based estimator, a **Residual Generator** (RG) and a **Decision Mechanism** (DM) and its main responsibility is to detect the faults.

The NN-estimator is trained in such a way so that the sensors (and actuators) signals are estimated and fed to the RG which then produces the residual using the estimated signals. Finally, the residuals are fed to the DM where a decision is made whether a sensor (and/or actuator) is faulty or not.

The estimator's inputs are obtained from the **Binary Switches** (BS) shown in Fig. 2b. This unit features three inputs: The first input represents the real measured values of \mathcal{U} and \mathcal{Y} , whereas the second one originates from the functions C_{u_j} and C_{y_j} defined as $C_{u_j} = [c_{u_1}, c_{u_2}, \dots, c_{u_{n_u}}]$ and $C_{y_j} = [c_{y_1}, c_{y_2}, \dots, c_{y_{n_y}}]$. In fact C_{u_j} and C_{y_j} represent two arrays that contain predefined functions, used during the training and operation phases of the *i*FD. The rationale behind choosing C_{u_j} and C_{y_j} is that having different value profiles compared to the values of the relevant sensor (and/or actuator) channel is useful to avoid *i*FD confusion during training. Designer experience, per application, target is advantageous in such case, i.e., to select initial value profile sets for C_{u_j} and C_{y_j} based on the known/expected sensor(s) (and/or actuator(s)) capabilities. Hence, a more realistic failure scenario can be provided to the *i*FD for training. We will illustrate this aspect in later sections of the paper. The third input is the **Isolation Signal**, IS_{y_j} , which is a binary input that controls the switching operation between its inputs e.g., from y_1 to c_{y_1}

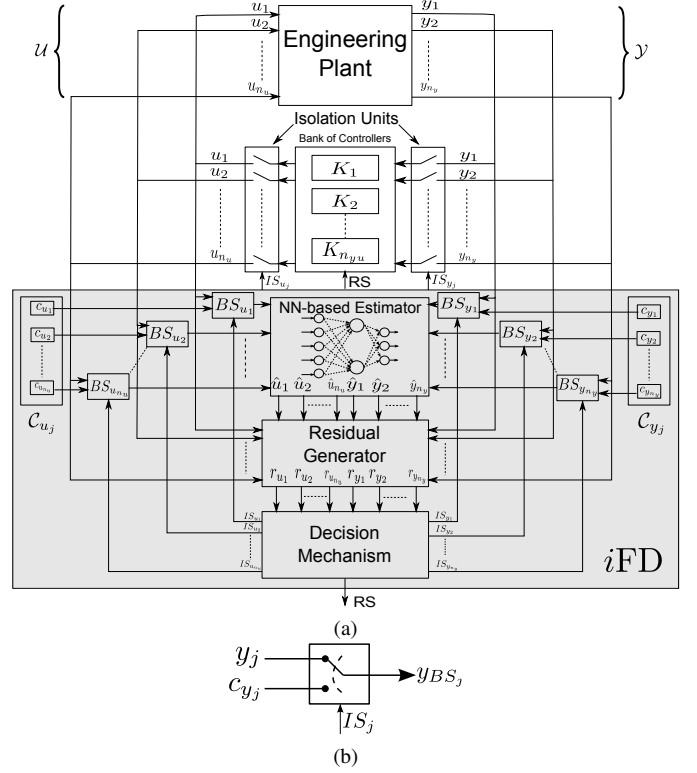


Fig. 2. (a) General diagram of actuator/sensor FDI with the proposed *i*FD, (b) Binary switch operation.

and vice versa. The output y_{BS_j} of the BS is given by,

$$y_{BS_j} = \begin{cases} y_j, & \text{if } IS_{y_j} = 1, \\ C_{y_j}, & \text{if } IS_{y_j} = 0. \end{cases} \quad (1)$$

The moving average filter defined in (1) generates the residual and accommodates the noise coming from the sensors, reducing in this way the FAR [17],

$$r_{y_j} = \sum_{j-(N-1)}^j \frac{(y_j - \hat{y}_j)^2}{N}, \quad (2)$$

where r_{y_j} is the residual, y_j and \hat{y}_j are the j^{th} real and estimated signals respectively (for the actuators the y is replaced by u), and N is the total number of past samples.

The DM decides whether one or more actuators and/or sensors are impaired or not by comparing each residual with a predefined threshold for each actuator/sensor, whereas the engineer has to pre-set the threshold for each residual, r_j . The **Reconfiguration Signal** RS at the output of DM essentially identifies a new controller when reconfiguration takes place. Both the IS_j and RS are switching signals and they change state when one or more faults (either actuator(s) or sensor(s)) is(are) detected. The IS_j isolates the faulty components from the loop and the RS adds a new controller in the loop which is designed to work with the remaining healthy components.

A. Offline training of the *i*FD unit: obtaining the learning set

The training procedure of the *i*FD can be done using one of the available training algorithms in the literature. The key point on the *i*FD functionality is the way the training data set is constructed. The *i*FD unit is trained with data accumulated from the various sub-sets of the main sensor set \mathcal{Y} . The collected data are then packed together in the structure format listed in Table I.

TABLE I
STRUCTURE OF THE DATA USED FOR THE *i*FD TRAINING.

Actuator/ Sensor Set Number	Actuator/ Sensor Status	Training data										Estimated sensor/actuators signals									
		y_1	y_2	y_3	\dots	y_{n_y}	u_1	u_2	u_3	\dots	u_{n_u}	\hat{y}_1	\hat{y}_2	\hat{y}_3	\dots	\hat{y}_{n_y}	\hat{u}_1	\hat{u}_2	\hat{u}_3	\dots	\hat{u}_{n_u}
1	Healthy sets:	$D_{y_1}^1$	$D_{y_2}^1$	$D_{y_3}^1$	\dots	$D_{y_{n_y}}^1$	$D_{u_1}^1$	$D_{u_2}^1$	$D_{u_3}^1$	\dots	$D_{u_{n_u}}^1$	$D_{\hat{y}_1}^1$	$D_{\hat{y}_2}^1$	$D_{\hat{y}_3}^1$	\dots	$D_{\hat{y}_{n_y}}^1$	$D_{\hat{u}_1}^1$	$D_{\hat{u}_2}^1$	$D_{\hat{u}_3}^1$	\dots	$D_{\hat{u}_{n_u}}^1$
	$\mathcal{Y} = y_1, y_2, \dots, y_{n_y}$	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	$\mathcal{U} = u_1, u_2, \dots, u_{n_u}$	$D_{y_1}^k$	$D_{y_2}^k$	$D_{y_3}^k$	\dots	$D_{y_{n_y}}^k$	$D_{u_1}^k$	$D_{u_2}^k$	$D_{u_3}^k$	\dots	$D_{u_{n_u}}^k$	$D_{\hat{y}_1}^k$	$D_{\hat{y}_2}^k$	$D_{\hat{y}_3}^k$	\dots	$D_{\hat{y}_{n_y}}^k$	$D_{\hat{u}_1}^k$	$D_{\hat{u}_2}^k$	$D_{\hat{u}_3}^k$	\dots	$D_{\hat{u}_{n_u}}^k$
2	Faulty set:	$C_{y_1}^2$	$D_{y_2}^2$	$D_{y_3}^2$	\dots	$D_{y_{n_y}}^2$	$D_{u_1}^2$	$D_{u_2}^2$	$D_{u_3}^2$	\dots	$D_{u_{n_u}}^2$	$C_{\hat{y}_1}^2$	$D_{\hat{y}_2}^2$	$D_{\hat{y}_3}^2$	\dots	$D_{\hat{y}_{n_y}}^2$	$D_{\hat{u}_1}^2$	$D_{\hat{u}_2}^2$	$D_{\hat{u}_3}^2$	\dots	$D_{\hat{u}_{n_u}}^2$
	y_1	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
		$C_{y_1}^k$	$D_{y_2}^k$	$D_{y_3}^k$	\dots	$D_{y_{n_y}}^k$	$D_{u_1}^k$	$D_{u_2}^k$	$D_{u_3}^k$	\dots	$D_{u_{n_u}}^k$	$C_{\hat{y}_1}^k$	$D_{\hat{y}_2}^k$	$D_{\hat{y}_3}^k$	\dots	$D_{\hat{y}_{n_y}}^k$	$D_{\hat{u}_1}^k$	$D_{\hat{u}_2}^k$	$D_{\hat{u}_3}^k$	\dots	$D_{\hat{u}_{n_u}}^k$
3	Faulty set:	$C_{y_1}^3$	$C_{y_2}^3$	$D_{y_3}^3$	\dots	$D_{y_{n_y}}^3$	$D_{u_1}^3$	$D_{u_2}^3$	$D_{u_3}^3$	\dots	$D_{u_{n_u}}^3$	$C_{\hat{y}_1}^3$	$C_{\hat{y}_2}^3$	$D_{\hat{y}_3}^3$	\dots	$D_{\hat{y}_{n_y}}^3$	$D_{\hat{u}_1}^3$	$D_{\hat{u}_2}^3$	$D_{\hat{u}_3}^3$	\dots	$D_{\hat{u}_{n_u}}^3$
	y_1, y_2	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
		$C_{y_1}^k$	$C_{y_2}^k$	$D_{y_3}^k$	\dots	$D_{y_{n_y}}^k$	$D_{u_1}^k$	$D_{u_2}^k$	$D_{u_3}^k$	\dots	$D_{u_{n_u}}^k$	$C_{\hat{y}_1}^k$	$C_{\hat{y}_2}^k$	$D_{\hat{y}_3}^k$	\dots	$D_{\hat{y}_{n_y}}^k$	$D_{\hat{u}_1}^k$	$D_{\hat{u}_2}^k$	$D_{\hat{u}_3}^k$	\dots	$D_{\hat{u}_{n_u}}^k$
4	Faulty set:	$C_{y_1}^4$	$C_{y_2}^4$	$D_{y_3}^4$	\dots	$D_{y_{n_y}}^4$	$C_{u_1}^4$	$D_{u_2}^4$	$D_{u_3}^4$	\dots	$D_{u_{n_u}}^4$	$C_{\hat{y}_1}^4$	$C_{\hat{y}_2}^4$	$D_{\hat{y}_3}^4$	\dots	$D_{\hat{y}_{n_y}}^4$	$C_{\hat{u}_1}^4$	$D_{\hat{u}_2}^4$	$D_{\hat{u}_3}^4$	\dots	$D_{\hat{u}_{n_u}}^4$
	y_1, y_2, u_1	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
		$C_{y_1}^k$	$C_{y_2}^k$	$D_{y_3}^k$	\dots	$D_{y_{n_y}}^k$	$C_{u_1}^k$	$D_{u_2}^k$	$D_{u_3}^k$	\dots	$D_{u_{n_u}}^k$	$C_{\hat{y}_1}^k$	$C_{\hat{y}_2}^k$	$D_{\hat{y}_3}^k$	\dots	$D_{\hat{y}_{n_y}}^k$	$C_{\hat{u}_1}^k$	$D_{\hat{u}_2}^k$	$D_{\hat{u}_3}^k$	\dots	$D_{\hat{u}_{n_u}}^k$
5	Faulty set:	$C_{y_1}^5$	$D_{y_2}^5$	$D_{y_3}^5$	\dots	$D_{y_{n_y}}^5$	$C_{u_1}^5$	$C_{u_2}^5$	$D_{u_3}^5$	\dots	$D_{u_{n_u}}^5$	$C_{\hat{y}_1}^5$	$D_{\hat{y}_2}^5$	$D_{\hat{y}_3}^5$	\dots	$D_{\hat{y}_{n_y}}^5$	$C_{\hat{u}_1}^5$	$C_{\hat{u}_2}^5$	$D_{\hat{u}_3}^5$	\dots	$D_{\hat{u}_{n_u}}^5$
	y_1, u_1, u_2	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
		$C_{y_1}^k$	$D_{y_2}^k$	$D_{y_3}^k$	\dots	$D_{y_{n_y}}^k$	$C_{u_1}^k$	$C_{u_2}^k$	$D_{u_3}^k$	\dots	$D_{u_{n_u}}^k$	$C_{\hat{y}_1}^k$	$D_{\hat{y}_2}^k$	$D_{\hat{y}_3}^k$	\dots	$D_{\hat{y}_{n_y}}^k$	$C_{\hat{u}_1}^k$	$C_{\hat{u}_2}^k$	$D_{\hat{u}_3}^k$	\dots	$D_{\hat{u}_{n_u}}^k$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
n_{yu}	Faulty sets:	$C_{y_1}^{n_{yu}}$	$C_{y_2}^{n_{yu}}$	\dots	$C_{y_{n_y-1}}^{n_{yu}}$	$D_{y_{n_y}}^{n_{yu}}$	$C_{u_1}^{n_{yu}}$	$C_{u_2}^{n_{yu}}$	\dots	$C_{u_{n_u-1}}^{n_{yu}}$	$D_{u_{n_u}}^{n_{yu}}$	$C_{\hat{y}_1}^{n_{yu}}$	$C_{\hat{y}_2}^{n_{yu}}$	\dots	$C_{\hat{y}_{n_y-1}}^{n_{yu}}$	$D_{\hat{y}_{n_y}}^{n_{yu}}$	$C_{\hat{u}_1}^{n_{yu}}$	$C_{\hat{u}_2}^{n_{yu}}$	\dots	$C_{\hat{u}_{n_u-1}}^{n_{yu}}$	$D_{\hat{u}_{n_u}}^{n_{yu}}$
	y_1, \dots, y_{n_y-1}	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	u_1, \dots, u_{n_u-1}	$C_{y_1}^{n_{yu}}$	$C_{y_2}^{n_{yu}}$	\dots	$C_{y_{n_y-1}}^{n_{yu}}$	$D_{y_{n_y}}^{n_{yu}}$	$C_{u_1}^{n_{yu}}$	$C_{u_2}^{n_{yu}}$	\dots	$C_{u_{n_u-1}}^{n_{yu}}$	$D_{u_{n_u}}^{n_{yu}}$	$C_{\hat{y}_1}^{n_{yu}}$	$C_{\hat{y}_2}^{n_{yu}}$	\dots	$C_{\hat{y}_{n_y-1}}^{n_{yu}}$	$D_{\hat{y}_{n_y}}^{n_{yu}}$	$C_{\hat{u}_1}^{n_{yu}}$	$C_{\hat{u}_2}^{n_{yu}}$	\dots	$C_{\hat{u}_{n_u-1}}^{n_{yu}}$	$D_{\hat{u}_{n_u}}^{n_{yu}}$

k is the total number of samples at each actuator/sensor set.

The first column identifies the sensor set number, $1 \dots n_{yu}$, defined as $n_{yu} = 2^{(n_u + n_y)} - 2^{n_u} - 2^{n_y} + 1$. The second column shows the status of the sensor set (all possible sensor/actuator fault scenarios are covered), and the next two columns are the training data (measured sensor and actuator signals).

The last two columns refer to the estimated sensors and actuators signals. The data set \mathcal{D} has dimensions $d_r \times d_c$ which are given by, $d_r = n_{yu} \times k$ and $d_c = n_y + n_u + n_{\hat{y}} + n_{\hat{u}}$. The number of estimated signals for actuators and sensors are noted $n_{\hat{u}}$ and $n_{\hat{y}}$ respectively. \mathcal{D} is constructed with data from appropriate simulations for every sensor/actuator fault scenario. Anywhere in the table, where the sensor(s) and or actuator(s) is(are) assumed to be faulty, a known function c_{u_j}, c_{y_j} replaces the k data points. For this reason, the design engineer needs to select a set of functions, C_{u_j} and C_{y_j} in order to replace the unpredicted outputs of the faulty actuators and sensors respectively. When an actuator and/or sensor fault occurs, the corresponding function, c_{u_j} and/or c_{y_j} is/are connected to the *i*FD. This is a result of the *i*FD's learning capability to respond to sensor/actuator faults in such a way where it continually checks for faults on the full sensor set and its sub-sets.

III. THE EMS SYSTEM - A TEST CASE

A. The EMS model

The single-stage, one degree-of-freedom model of the EMS system represents a quarter of a typical maglev vehicle. A rigorous analysis on the non-linear model and the linearization is discussed in [15]. The EMS is a non-linear, inherently unstable, and critical-safety system with non-trivial control requirements. The EMS consists of an electromagnet with a ferromagnetic core and a coil of N_c turns which is attracted to the rail track (made of ferromagnetic material). The vehicle mass, M_s is supported on the electromagnet, with z_t being

the rail track's position, z the electromagnet's position (both are accounted for small variations around the operating point), and $(z_t - z)$ is the distance between them, namely the airgap. The airgap is to be kept as close as possible to the operating point at a range that will not infringe the maximum allowed as shown later in this section. The non-linear model is given by,

$$\begin{aligned} \frac{dI}{dt} &= \frac{V_c - IR_c + \frac{N_c A_p K_b}{G^2} \left(\frac{dz_t}{dt} - \frac{dZ}{dt} \right)}{\frac{N_c A_p K_b}{G} + L_c}, F = K_f B^2, \\ \frac{d^2 Z}{dt^2} &= g - \frac{K_f I^2}{M_s G^2}, B = K_b \frac{I}{G}, \frac{dG}{dt} = \frac{dz_t}{dt} - \frac{dZ}{dt}, \end{aligned} \quad (3)$$

where V_c is the coil's voltage, F is the vertical force, I is the coil's current, G is the airgap, Z is the electromagnet's position, B is the flux density, M_s is the vehicle's mass, R_c is the coil's resistance, L_c is the coil's inductance, N_c is the number of turns and A_p is the pole face area. K_b , K_f and g reflect the flux, force and gravity constants with values equal to $0.0015 Tm/A$, $0.0221 Nm^2/A^2$ and $9.81 m/s^2$ respectively.

Although linearization of the EMS is needed for the design of the linear controllers, in this work the controllers are tuned and tested with the nonlinear model of the EMS. Linearization is performed on small variations around the operating point, i.e., the linearized model for the airgap is given by $G = G_o + (z_t - z)$, where lower case letters characterize small variations, and subscript 'o' refers to the operating point. The same notation is used for B , F , I , V_c and Z variables.

Following [15], a linear time invariant state space model of the EMS is expressed as $\dot{x} = Ax + B_{u_c} u_c + B_{z_t} \dot{z}_t$ and $y = Cx$, where A is the state matrix, B_{u_c} is the control input matrix, B_{z_t} is the disturbance input matrix and C is the output matrix. The state vector is given by $x = [i \quad \dot{z} \quad (z_t - z)]^T$, where i is the current, \dot{z} and $(z_t - z)$ are the vertical velocity and the airgap to be controlled, \dot{z}_t is the velocity of the rail

track input and u_c is the control input (driving voltage). The state space matrices for the operating point with a vehicle mass of $M_s = 1000\text{kg}$ and the parameters of the coils are given in [14].

B. Disturbance inputs and control requirements of the EMS

Two input disturbances are considered at the vertical axis of the EMS system. Typically, the EMS must be able to follow the gradient onto the rail (i.e., Deterministic (Dtm) input) and reject the disturbances caused by rail track irregularities (i.e., Stochastic (Sth) input). The Deterministic input is driven by the position change of the rail, z_t , caused by the intended changes of the rail track's gradient while the vehicle is moving along the rails. Dtm input is assumed a 5% rail track gradient at a vehicle speed of 15m/s with an acceleration of 0.5m/s^2 and a jerk of 1m/s^3 . The control requirements of the EMS system are: maximum airgap deviation, $(z_t - z)_p \leq 7.5\text{mm}$, maximum control effort, $u_{cp} \leq 300\text{V}$, settling time, $t_s \leq 3\text{s}$, and airgap steady state error, $e_{(z_t - z)ss} = 0$. The Stochastic input is driven by the random variations of the rail track with the velocity variations approximated by a double-sided power spectrum density [14]. The control requirements for the stochastic case are: **Root Mean Square (RMS)** of acceleration, $\ddot{z}_{rms} \leq 1\text{m/s}^2$, **RMS** of airgap variation, $(z_t - z)_{rms} \leq 5\text{mm}$, **RMS** of control effort, $u_{c,rms} \leq 300\text{V}$.

IV. EFFICACY AND ASSESSMENT OF THE PROPOSED iFD

A. The iFD applied on the EMS system

The EMS system has one input (i.e., $n_u = 1$), $\mathcal{U} = \{u_c\}$, and four possible output measurements (i.e., $n_y = 4$), $\mathcal{Y} = \{i, (z_t - z), \dot{z}, \ddot{z}\}$, that can be used for controller design. The \mathcal{H}_∞ Loop-Shaping Design Procedure (LSDP) robust control method is used to achieve the necessary closed-loop performance, which requires to have the controlled variable (i.e., $z_t - z$) as a standard input to the controller [18]. Hence, three sensor fault possibilities are assumed for the EMS, i.e., i , \dot{z} and \ddot{z} . Since $(z_t - z)$ is a standard output measurement, the total number of actuator/sensor sets is, $n_{yu} = 8$. Consequently, eight controllers of this type (i.e., $K_{(z_t - z)}$, $K_{i, (z_t - z)}$, \dots , etc.) are utilized to cover all seven possible sensor fault combinations as illustrated in Fig. 3a.

B. Neural Network estimator offline training

A dynamic nonlinear input-output NN model with tapped delay lines at the input was used for time-series prediction. The NN algorithm is adapted to fit the inputs and targets of the EMS for the iFD realization. It has a total of five inputs (u_c and $i, (z_t - z), \dot{z}, \ddot{z}$) and three estimated outputs, $(\hat{i}, \hat{\dot{z}}, \hat{\ddot{z}})$ and is realized as a hidden layer (with one delay and 20 hidden neurons) and an output layer with sigmoid and linear functions respectively. A fast convergence method used for training moderate sized feed-forward neural networks is the Levenberg-Marquardt backpropagation algorithm [19]. Two stopping criteria were set as: i) the **Mean Square Error** ($\text{MSE} \leq 10^{-5}$) or, ii) a maximum of 1000 epochs. The total time for the training process is about 30 min on a typical desktop PC (Intel i3@2GHz, 8GB RAM).

The EMS has one actuator $n_u = 1$ and one estimated actuator signal $n_{\hat{u}} = 1$. Moreover, the number of sensors is $n_y = 4$, the estimated sensor signals number $n_{\hat{y}} = 3$ (air gap is

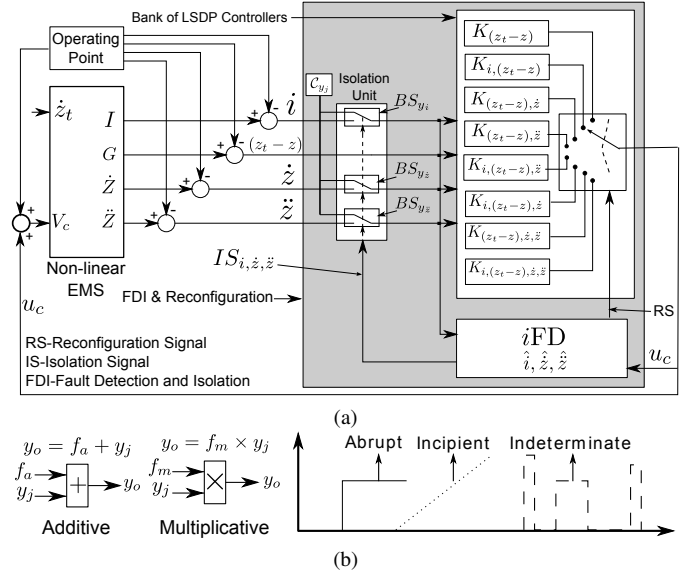


Fig. 3. (a) The proposed iFD applied to the EMS system, (b) Sensor fault categories and types.

assumed to be faultless hence is not included), and the number of sensor (actuator) sets $n_{yu} = 8$. The training data from each sensor set that composes \mathcal{D} were collected at a sample rate of $\tau_s = 1\text{kHz}$ with a total simulation time of $T = 6.6\text{s}$. The data set \mathcal{D} consists of the data sets \mathcal{D}_d and \mathcal{D}_s , drawn from the Dtm and Sth responses of the EMS¹. The data set used for training is given by $\mathcal{D}^{d_r \times d_c} = \begin{bmatrix} \mathcal{D}_d^{d_{r_d} \times d_{c_d}} & \mathcal{D}_s^{d_{r_s} \times d_{c_s}} \end{bmatrix}^T$ where, $d_r = d_{r_d} + d_{r_s}$ and $d_c = d_{c_d} + d_{c_s}$ are calculated as explained in Section II-A. The total number of columns is calculated as $d_c = 9$, hence for a total number of samples per set $k = 6.6 \times 1000$, the data set sizes $d_{r_d} = d_{r_s} = 52800$ and $d_r = 105600$. The functions used for the training of the NN are $\mathcal{C}_{y_j} = \{c_i = 0, c_z = 0, c_{\dot{z}} = 0\}$ with dimensions $k \times 3$. These functions are both used in the Dtm and Sth responses of the EMS where a sensor fault occurs, as explained in Section II-A.

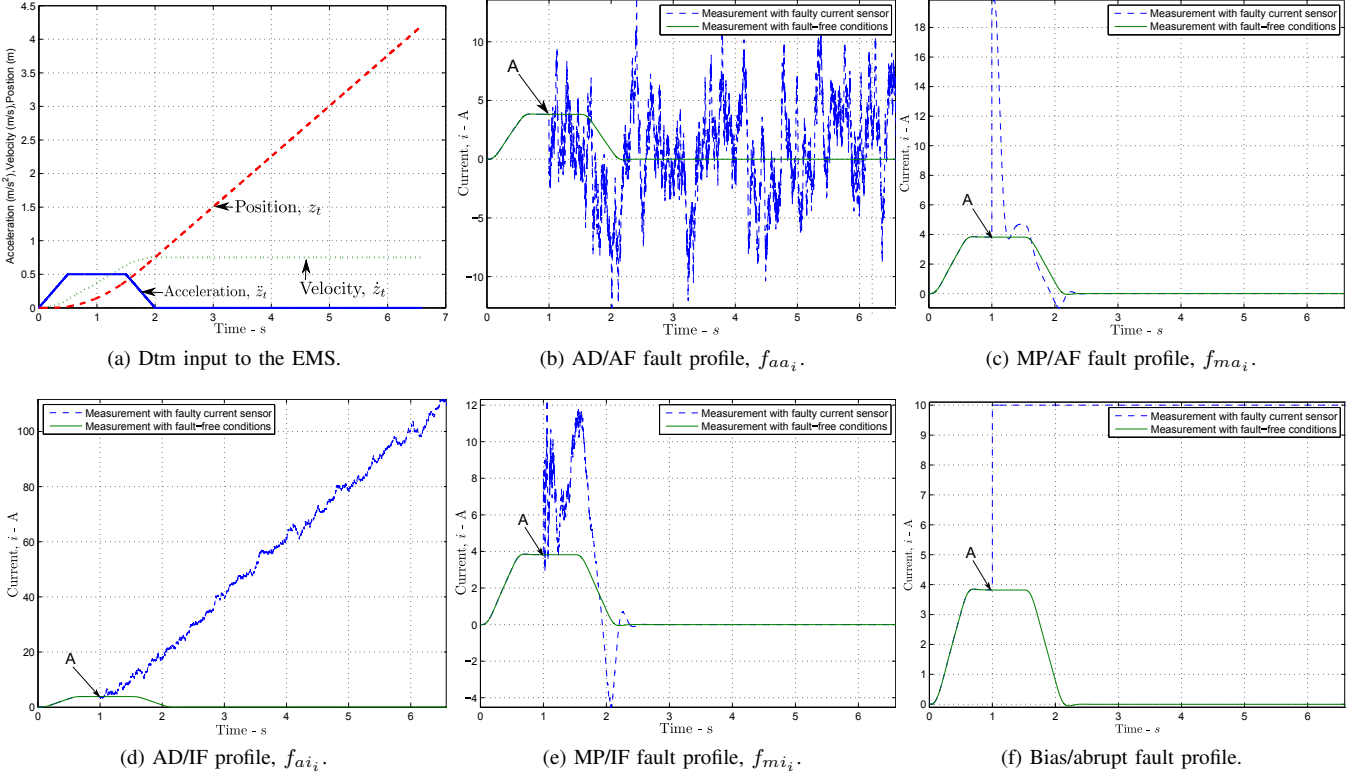
C. Sensor fault scenarios for the EMS

Sensor faults are categorized into **AD** and **MP** (see Fig. 3b). AD and MP categories are both considered for the sensor fault scenarios i.e., the output of the sensor, y_j , is added or multiplied with a function f_a and f_m respectively. Three types of faults exist in each category were set as follows: i) **Abrupt Fault (AF)**: when an actuator/sensor component is damaged and its output changes in a step-wise fashion; this fault can be rough but easier to detect and therefore isolated from the control loop, ii) **Incipient Fault (IF)**: this type of fault develops slower, hence it is harder to detect. It is generally very dangerous because it could destabilize the control loop before it is detected, iii) **Indeterminate Fault (InF)**: it can have characteristics similar to the AF and IF but it occurs in random width time windows and amplitudes. This fault is equally dangerous like, in ii), because it appears and vanishes at any time without warning. This type of fault is treated like PF by the iFD, and is 'captured' once it appears.

In total, there are 4 sensors in the full sensor set, \mathcal{Y} , with the assumption that the airgap sensor ($z_t - z$) cannot fail² (also

¹Subscripts d and s indicate the Dtm and Sth cases respectively.

²In case where fault tolerance is required for the airgap then redundant components can be used under a voting scheme.

Fig. 4. Coil's current sensor, i , fault profiles.

note that the control actuator u_c is assumed not to fail). The AD, MP categories and bias faults are utilized for each one of the i , \dot{z} and \ddot{z} sensors.

The fault profiles used for each sensor are similar to the ones illustrated for the current sensor, i , in Fig. 4b-4f. In all cases the faults start developing at one second, i.e., the sensor fault time $t_f = 1s$, is marked at point A in all figures. The Dtm input to the EMS is illustrated in Fig. 4a. Fig. 4b for the impaired sensor illustrates the normal current value superimposed with a low frequency band-limited random signal, $\nu(t)$, at frequencies of $10rad/s$ and zero mean, white noise characteristics and power spectral density of $S_i = 5$ limited to $\omega_i = 1.6Hz$. As a consequence, the AD/AF profile for the current sensor, f_{aa_i} is given by (4). The same pattern is used for the \dot{z} and \ddot{z} sensors with the same bandwidth and $S_{\dot{z}} = 0.03$ and $S_{\ddot{z}} = 2$ respectively. Fig. 4c depicts the MP/AF, f_{ma_i} , where the current sensor suddenly fails at $t_f = 1s$ and as a result its output becomes five times larger than normal (4). Similar fault profile is used in the case of \dot{z} and \ddot{z} .

$$f_{aa_i}(t) = \begin{cases} \nu(t), & \text{if } t_f \leq t < \infty, \\ 0, & \text{if } t < t_f, \end{cases} \quad (4)$$

$$f_{ma_i}(t) = \begin{cases} 5, & \text{if } t_f \leq t < \infty, \\ 1, & \text{if } 0 \leq t < t_f. \end{cases}$$

The bias/abrupt type of fault for the current measurement is shown in Fig. 4f. It is clearly shown that the output of the sensor abruptly increases up to its maximum value, where for the current sensor case is $10A$, i.e., $y_{oi} = 10A$.

IFs are illustrated in Fig. 4d and Fig. 4e respectively. In the former figure the AD/IF on the current measurement, f_{ai_i} are described by (5). The latter is a ramp type signal with σ_i slope superimposed with a low frequency random signal with

band-limited white noise characteristics as it was previously explained. The aforementioned figure, depicts the MP/IF, f_{mi_i} described by (5), where the fault starts developing at $t_f = 1s$ and then falls to zero (due to multiplication by zero).

$$f_{ai_i}(t) = \begin{cases} \sigma_i(t - t_f) + \nu(t), & \text{if } t_f \leq t < \infty, \\ 0, & \text{if } 0 < t < t_f, \end{cases} \quad (5)$$

$$f_{mi_i}(t) = \begin{cases} \sigma_i(t - t_f) + 1 + \nu(t), & \text{if } t_f \leq t < \infty, \\ 1, & \text{if } 0 < t < t_f. \end{cases}$$

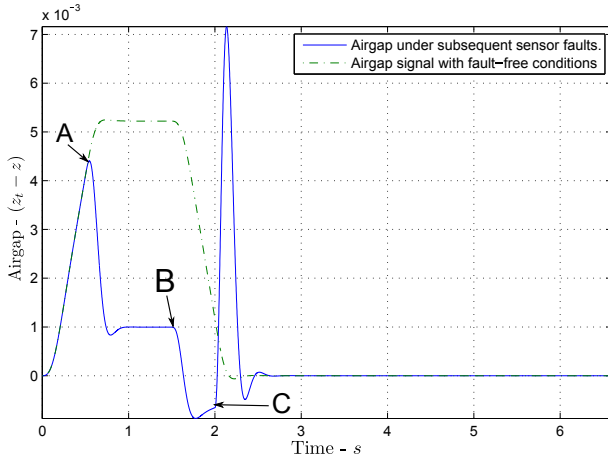
The fault profiles for \dot{z} and \ddot{z} follow the same behaviour as above with effective slopes, $\sigma_{\dot{z}} = 20$, $\sigma_z = 6$ and $\sigma_{\ddot{z}} = 20$ and for the PSD, $S_i = 0.5$, $S_{\dot{z}} = 0.03$ and $S_{\ddot{z}} = 0.5$ respectively.

The following scenario is used for the detailed explanation of the i FD working principle: i) the Dtm disturbance is applied to EMS, ii) three sensors are subsequently impaired with a time difference as follows: \ddot{z} at $0.5s$, \dot{z} at $1.5s$ and i at $2s$, iii) a MP/AF profile is used for each sensor e.g., for i see Fig. 4c.

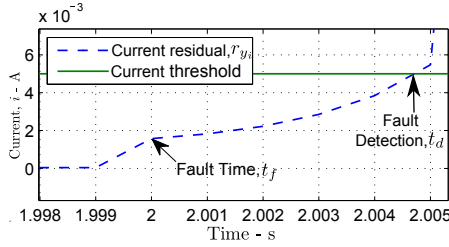
Fig. 5a illustrates the airgap with the fault-free case (i.e., healthy sensor set, \mathcal{Y} , with $K_{i,(z_t-z),\dot{z},\ddot{z}}$) and under the fault scenario mentioned. The \ddot{z} sensor is impaired at $0.5s$ (designated at point A) and immediately after a controller reconfiguration follows i.e., a new controller, $K_{i,(z_t-z),\dot{z}}$, is introduced in the loop) to maintain the performance of the EMS³. Next, \dot{z} fails at $t_f = 1.5s$ (point B) and i follows at $t_f = 2s$ (point C). The subsequent faults are successfully detected and accommodated via appropriate switching on $K_{i,(z_t-z)}$ and $K_{(z_t-z)}$ respectively.

Taking the i (current sensor) as example, the sensor fault

³Note that the EMS response in both cases comply with the requirements described in Section III.



(a) Airgap sensor signal.

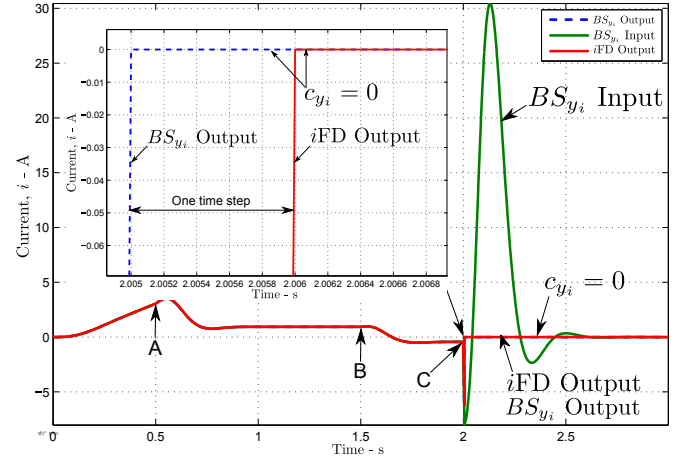
(b) Current sensor residual, r_{y_i} .Fig. 5. Fault scenario with $id : 8$ in Table II.

accommodation for each failure is integrated in three steps: i) *Sensor FD*: when the fault occurs at $t_f = 1s$, the residual of the current measurement, r_{y_i} starts increasing and as soon as it passes the threshold (see Fig. 5b) the fault is detected, ii) *Fault Isolation*: At this stage the faulty sensor is removed from the loop using a BS, and $c_{y_i} = 0$ is connected at the input of the iFD . Fig. 6 shows the signal at the input/output of the BS and the signal at the output of the iFD , iii) *Controller reconfiguration*: a reconfiguration signal is generated and the new controller, $K_{(z_t - z)}$, is introduced in the loop.

A close inspection in Fig. 6, after the fault occurs at point C, indicates that it takes a few time steps to detect the fault for the iFD and one time step for the BS_{y_i} to permanently change its output to $c_{y_i} = 0$. As a result, the residual is always large, which effectively justifies why with the proposed method the output of the BS will never return to its previous stage when or if the fault vanishes. The input to BS_{y_i} with the two previous sensor faults, more specifically the acceleration and velocity, (at point A and B respectively) is also shown.

Table II gives the resulted performance of the EMS and the FA after 35 faults for each Dtm and Sth responses of the EMS. The first column of the table describes the sensor fault scenarios used to test the proposed iFD . Typically, rows 2-4 show the results for single sensor faults that occur at $t_f = 1s$, whereas the rest of the rows show the results with subsequent faults starting from $0.5s$ with a time difference of $1s$. The first six columns present the performance with AFs, whereas the rest four show the performance with IFs. The EMS was tested with the Dtm and Sth disturbance inputs as given in Section III-B. MP and AD faults are used, as well as a bias (Bis.) fault that occurs abruptly. In each scenario, if the performance of the EMS is successfully maintained is indicated with \checkmark or otherwise with \times . Also, if an FA occurs is marked with \checkmark .

A close inspection in the table shows that the iFD suc-

Fig. 6. Input-output of the BS_{y_i} of the current sensor, i , and output of the iFD .TABLE II
PERFORMANCE WITH VARIOUS SENSOR FAULT SCENARIOS FOR THE EMS.

id	Faulty sensor(s)	AF						IF			
		MP/FA		AD/FA		Bis./FA		MP/FA		AD/FA	
		Sth.	Dtm.	Sth.	Dtm.	Sth.	Dtm.	Sth.	Dtm.	Sth.	Dtm.
1	Fault-free	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
2	i	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
3	\ddot{z}	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
4	\ddot{z}	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
5	$i \rightarrow \ddot{z}$	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
6	$i \rightarrow \ddot{z}$	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
7	$\ddot{z} \rightarrow \ddot{z}$	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
8	$\ddot{z} \rightarrow \ddot{z} \rightarrow i$	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark

cessfully detects and reconfigures the controller in every case, and performance is successfully maintained in all scenarios even though, in some cases, an FA appears. The FA, which appears in certain cases in $id : 7 - 8$, means that a sensor looks impaired although it is actually healthy. Substantial conclusions regarding occurred FA have arisen during various simulations: i) the set threshold for the residuals plays a vital role: If for example a sensor, y_1 is impaired, then during the transition from one controller to another the residuals of the healthy sensors may increase as well, and if the residuals thresholds are low enough a FA will occur. If the residual threshold value of the other sensors is increased and under an assumed situation where these sensors also develop a fault then FD delays could drive the system to instability⁴, ii) the ability of the NN nature of the iFD to estimate the faults causes delay in the detection of a fault, since it generally leaves a small residual for some time after a fault occurs [13], iii) there is coupling between the integrated units in the reconfigurable FTC scheme. The NN estimator, the residual generator, the decision and reconfiguration mechanisms are strongly interconnected in a way that a small parameter change in one may affect the performance of the other.

Sensitivity issues due to the aforementioned interactions form current research which authors are investigating.

Table III outlines the FD time for each scenario including the detection time in FA. The first column is the identifier number for the selected scenario and the second column lists the sensors (the underlined ones indicate those to be impaired

⁴Threshold levels are calculated via trial-and-error in this case to avoid sensitivity to noise and allow for appropriate fault detection.

TABLE III
FD TIME USING THE *i*FD WITH VARIOUS SENSOR FAULT SCENARIOS.

id	Sens.	t_f (s)	Fault detection time, t_d (s)							
			AF				IF			
			Mult.	Dtm.	Sth.	Dtm.	Mult.	Dtm.	Sth.	Dtm.
1	\hat{z}	1.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
2	\hat{z}	1.0	1.000	1.000	1.000	1.000	1.000	1.000	2.399	1.000
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
3	\hat{z}	1.0	1.000	1.000	1.000	1.000	1.000	1.000	1.030	1.000
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
4	\hat{z}	0.5	0.500	0.500	0.500	0.500	0.500	0.500	0.527	0.500
	\hat{z}_t	1.5	1.500	1.500	1.500	1.500	1.500	1.500	2.335	1.511
	\hat{z}_t	-	-	-	-	-	-	-	-	-
5	\hat{z}	0.5	0.500	0.500	0.500	0.500	0.500	0.500	0.527	0.500
	\hat{z}_t	1.5	1.500	1.500	1.500	1.500	1.500	1.500	1.510	1.514
	\hat{z}_t	-	-	-	-	-	-	-	-	-
6	\hat{z}	0.5	0.500	0.500	0.500	0.500	0.500	0.500	1.700*	1.700*
	\hat{z}_t	1.5	1.500	1.500	1.500	1.500	1.500	1.500	2.403	1.500
	\hat{z}_t	-	-	-	-	-	-	-	-	-
7	\hat{z}	2.0	2.000	0.654*	2.000	0.654*	2.000	0.650*	2.614	0.654*
	\hat{z}_t	1.5	1.517	1.502	1.500	1.500	1.500	1.500	1.517	1.502
	\hat{z}_t	0.5	0.506	0.500	0.500	0.500	0.500	0.500	0.500	0.500

at t_f as shown in the next column). The rest of the values account for the FD time t_d . The values marked in bold show that the occurred fault is detected and in bold* indicate the false alarms.

A careful observation of the results shows that with abrupt single sensor failures ($id : 1-3$) the fault detection is achieved just when the sensor fails, but in the IF cases, the fault is detected with a certain delay in between $0.030s - 0.399s$. However, this delay causes no problems on the stability or performance of the EMS before or after the controller reconfiguration. Although delays are observed in FD through these three scenarios and the next two, none of them causes FAs. The last two scenarios (in the sixth two subsequent faults on velocity and acceleration measurements are considered and in the seventh the acceleration, velocity and current sensors are impaired sequentially) FAs appear mainly on the current sensor. It should be mentioned that other observed fault delays in *i*FD are successfully accommodated through the controller reconfiguration.

D. Comparison with a bank of KEs

In this section a comparison is done between the *i*FD unit and a typical model-based FD unit composed of a bank of linear KEs. A typical bank of KEs (\mathcal{O}_{y_i}) is shown in Fig. 7. The inputs to the FD are the full sensor set \mathcal{Y} and the control input u_c of the EMS system. For more details regarding the design of such an FD unit the reader can refer to [15]. The comparison is done at the simulation level using the same fault scenarios described in Section IV-C. The fault detection time, t_d from each sensor fault scenario is tabulated in Table IV. By comparing the t_d values with the ones in Table III it can be seen that the model-based approach has smaller detection time. This occurs because the *i*FD unit is using a NN estimator which can estimate the fault itself, therefore the development of the residual is relatively slow. The results clearly show that the *i*FD can successfully detect the sensor faults before the performance/stability of the EMS is seriously affected. It should also be noted that the model-based approach requires the model being precise enough. This is true in this case for the EMS model, however in many complex engineering applications it will not be the case. Nonlinearities and model uncertainty would impact modeling and the NN approach for FD will be seen a more favourable approach to take.

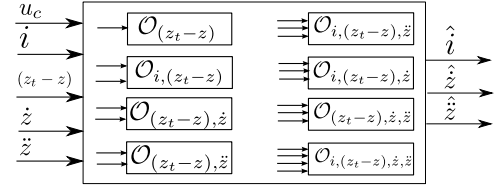


Fig. 7. Conventional approach for multiple sensor failure detection with KEs.

TABLE IV
FD TIME USING A BANK OF KES WITH VARIOUS SENSOR FAULT SCENARIOS.

id	Sens.	t_f (s)	Fault detection time, t_d (s)							
			AF				IF			
			Mult.	Dtm.	Sth.	Dtm.	Mult.	Dtm.	Sth.	Dtm.
1	\hat{z}	1.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
2	\hat{z}	1.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
3	\hat{z}	1.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	\hat{z}_t	-	-	-	-	-	-	-	-	-
	\hat{z}_t	-	-	-	-	-	-	-	-	-
4	\hat{z}	0.5	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	\hat{z}_t	1.5	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
	\hat{z}_t	-	-	-	-	-	-	-	-	-
5	\hat{z}	0.5	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	\hat{z}_t	1.5	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
	\hat{z}_t	-	-	-	-	-	-	-	-	-
6	\hat{z}	0.5	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	\hat{z}_t	1.5	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
	\hat{z}_t	-	-	-	-	-	-	-	-	-
7	\hat{z}	2.0	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	\hat{z}_t	1.5	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
	\hat{z}_t	0.5	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50

E. Comparison of the execution time

A comparison of the simulation elapsed time for the bank of KEs (shown in Fig. 7) and the *i*FD is done using the full sensor set \mathcal{Y} . The execution time, t_e is measured over a high level simulation in Simulink platform iteratively for 100 times. The execution time for each simulation is depicted in Fig. 8. The average simulation time for the *i*FD is about $0.5s$ and for the bank of KE about $8s$. Fig. 8 clearly shows that the *i*FD is about 16 times faster than the bank-estimator approach and proves that the proposed fault detection method is adequately fast and worth considering for detecting actuators/sensors faults in real industrial applications.

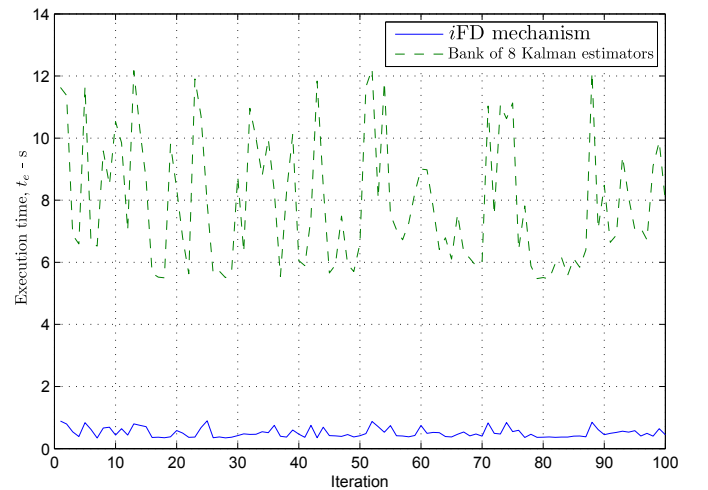


Fig. 8. Execution time of *i*FD unit vs. bank of Kalman estimators.

V. DISCUSSION AND CONCLUSIONS

A. Discussion

It has been demonstrated that a single NN estimator can replace a bank of estimators and is able to detect multiple faults with the advantage of having lower computational cost and simplified programming. Based on the simulation results some points are discussed in this section before the conclusion:

- i) IFs are considered as PFs because once they are detected by the i FD they are permanently removed from the loop,
- ii) if a component appears to be faulty (false alarm) then it is permanently removed from the loop and never inserted back even if the fault vanishes,
- iii) collecting the training data is a time consuming process since the controller has to be designed for each actuator/sensor set, but nevertheless the training of the i FD requires a short time and is done only once,
- iv) there is no analytical method for the user to define c_j for each actuator and sensor, but the rationale behind it is that its values should never be the same as the actuator/sensor otherwise the i FD will be confused,
- v) threshold selection is a non-trivial task to perform, since on one hand it affects the fault detection sensitivity of the DM and on the other it directly affects the FAR. If the DM is very sensitive to faults (thresholds are set too low) then the FAR increases, or if it is insensitive (high thresholds), the loop may actually go unstable before the reconfiguration happens. Taking into account the property of the i FD's AI nature which is able to estimate the faults, the threshold selection needs to be done very carefully with the trial and error method [20]. Although this field has gain a lot of attention is beyond the scope of this paper. Threshold selection for example is also related to the nature of the faults. Since soft faults leave small residuals, the threshold is set lower but this can increase the FA rate. On the contrary, hard faults leave large residual, therefore threshold is set higher. Another solution found in the literature is the adaptive threshold approach which improves robust detection of faults [21],
- vi) the advantage of i FD in terms of reducing complexity and computational expense is clear, while the model-free nature of NNs helps on dealing with model approximation. KEs work well regarding the avoidance of false alarms and no delay in fault detection, however require good model knowledge and at expense of complexity (bank of KEs).

B. Conclusions

A detailed analysis of an FD scheme applied on an EMS with 70 fault scenarios is presented. The tests accounted for scenarios with a number of typical sensor fault characteristics. The results show that a single NN estimator scheme can be used in the FDI instead of multiple which has less complexity and computational resources. A few simple modifications in the proposed architecture allow the replacement of the bank of estimators with a typical NN. The results show that this new approach has strong potential to replace multiple estimators used in FDI schemes in industrial applications. Robustness properties and false alarm rates of the proposed scheme is work that the authors investigate currently.

ACKNOWLEDGMENT

The authors would like to thank Prof. R.M. Goodall from School of Electronic, Electrical and Systems Engineering, Loughborough University, UK for his advices on the EMS and the anonymous reviewers for their valuable comments and suggestions that improved the quality of this paper.

REFERENCES

- [1] M. R. Napolitano, D.A. Windon, J.L. Casanova, M. Innocenti, and G. Silvestri. Kalman filters and neural-network schemes for sensor validation in flight control systems. *IEEE Transactions on Control Systems Technology*, 6(5):596–611, 1998.
- [2] M. Ranjbaran and K. Khorasani. Fault recovery of an under-actuated quadrotor aerial vehicle. In *49th IEEE Conference on Decision and Control*, pages 4385–4392, 2010.
- [3] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag NY, 2003.
- [4] I. Hwang, S. Kim, Y. Kim, and C. E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, 2010.
- [5] G. Campa, M. L. Fravolini, B. Seanor, M. R. Napolitano, D. D. Gobbo, G. Yu, and S. Gururajan. On-line learning neural networks for sensor validation for the flight control system of a B777 research scale model. *International Journal of Robust and Nonlinear Control*, 12(11):987–1007, 2002.
- [6] H. A. Talebi and K. Khorasani. A neural network-based actuator gain fault detection and isolation strategy for nonlinear systems. In *46th IEEE Conference on Decision and Control*, pages 2614–2619, 2007.
- [7] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani. Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics*, 18(2):90–99, 2008.
- [8] I. Samy, I. Postlethwaite and D-W. Gu. A comparative study of NN- and EKF-based SFDA schemes with application to a nonlinear UAV model. *International Journal of Control*, 83(5):1025–1043, 2010.
- [9] I. Samy, I. Postlethwaite and D-W. Gu. Survey and application of sensor fault detection and isolation schemes. *Control Engineering Practice*, 19(7):658–674, 2011.
- [10] E. N. Skoundrianos and S. G. Tzafestas. Modelling and FDI of dynamic discrete time systems using a MLP with a new sigmoidal activation function. *Journal of Intelligent and Robotic Systems*, 41(1):19–36, 2004.
- [11] R. Isermann. Supervision, fault-detection and fault-diagnosis methods - an introduction. *Control Engineering Practice*, 5(5):639–652, 1997.
- [12] V. Reppa, M. M. Polycarpou, and C. G. Panayiotou. Adaptive approximation for multiple sensor fault detection and isolation of nonlinear uncertain systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):137–153, 2014.
- [13] K. Michail and K. M. Deliparaschos. Sensor fault detection with low computational cost: A proposed neural network-based control scheme. In *17th IEEE Conference on Emerging Technologies and Factory Automation*, 2012.
- [14] K. Michail, K. M. Deliparaschos, S. G. Tzafestas, and A. C. Zolotas. AI-based low computational power Actuator/Sensor fault detection applied on a MAGLEV suspension. In *21st Mediterranean Conference on Control & Automation (MED)*, pages 1127–1132, 2013.
- [15] K. Michail. *Optimised Configuration of Sensing Elements For Control And Fault Tolerance Applied To An Electro-Magnetic Suspension System*. PhD thesis, 2009. Loughborough University, School of Electronic, Electrical and Systems Engineering. <http://hdl.handle.net/2134/5806>.
- [16] P. Humpherys, J. Redd and J. West. A fresh look at the kalman filter. *SIAM review*, 54(4):801–823, 2012.
- [17] G. Heredia, A. Ollero, R. Mahtani, M. Bjar, V. Remuss, and M. Musial. Detection of sensor faults in autonomous helicopters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2229–2234, 2005.
- [18] D. C. McFarlane and K. Glover. A loop-shaping design procedure using \mathcal{H}_∞ synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, 1992.
- [19] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.
- [20] Z. Wang and C. Lu. An adaptive threshold based on RBF neural network for fault detection of a nonlinear system. In *Advances in Computer, Communication, Control and Automation*, v.121, pages 495–502, Springer Berlin Heidelberg, 2012.
- [21] V. Montes de Oca, S. Puig and J. Blesa. Robust fault detection based on adaptive threshold generation using interval LPV observers. *International Journal of Adaptive Control and Signal Processing*, 26(3):258–283, 2012.